

Faculty of Information Technology

*Computer Science Department*

*Computer Science*

Urban Parking Solutions Mobile Application”ParkEase”

Graduation Project 1 Report

Prepared by

Supervised By

Dr. Adi Maaitah

| Student ID | Student Name |
| --- | --- |
| 202020345 | Mais Zahran |

To Obtain

**BSc** In Computer Science

1ST Semester / 2023/2024

Group No : CS-23-1-1-Serial(Replace serial by group serial number)

Copyright © 2023-2024 - All rights Reserved

Middle East University

**Declaration**

We hereby acknowledge that the work presented in this document report and the ideas based upon are the group members own unless stated otherwise and properly cited in the text and referenced at the end of the document.

| Student ID | Student Name | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Supervisor Approval موافقة المشرف** |
| Approval For Submission  I certify that this project report entitled “**Urban Parking Solutions Mobile Application”ParkEase**” was prepared by **Mais Zahran** has met the required standard fro submission in partial fulfillment of the requirements for the degree of Bachelor of science in Computer Science at MEU  Approved by  Signature : ……………………………………………………..  Supervisor : Dr ………………………………………………  Date:……………………………………………………………… |
|  |

|  |
| --- |
| **Acknowledgements الشكر و العرفان** |
| **ACKNOWLEDGEMENTS**  I would like to thank everyone who had contributed to the successful completion of this project starting from the guidance of our supervisors  Dr.Adi Maaitah and the team work with my colleagues who worked with this project. |
|  |

|  |
| --- |
| **Abstract (English) المستخلص (إنجليزي)** |
| Urban Parking Solutions Mobile Application”ParkEase”  This project introduces a mobile application designed to revolutionize urban parking, with a special emphasis on supporting electric vehicles (EVs). As cities grow and EV adoption increases, finding parking and charging stations becomes a significant challenge. This application addresses these issues by offering real-time information on available parking spaces and specific functionalities for EV owners, such as locating EV charging stations and facilitating reservation and payment for parking and charging. Employing Agile methodologies, the project aims to enhance urban mobility, promote environmental sustainability by encouraging EV use, and contribute to reducing urban congestion. Through innovative technology, it sets a new standard for smart, EV-friendly urban parking solutions, simplifying the parking experience in the modern urban landscape. |
|  |

|  |
| --- |
| **Abstract (Arabic) المستخلص (عربي)** |
| Urban Parking Solutions Mobile Application”ParkEase”  يقدم هذا المشروع تطبيقًا جوالًا مصممًا لتحويل تجربة البحث عن مواقف في المناطق الحضرية، مع تركيز خاص على دعم السيارات الكهربائية (EVs). مع نمو المدن وزيادة تبني السيارات الكهربائية، يصبح إيجاد مواقف ومحطات شحن تحديًا كبيرًا. يتناول هذا التطبيق هذه المشكلات من خلال تقديم معلومات فورية عن أماكن الركن المتاحة ووظائف محددة لمالكي السيارات الكهربائية، مثل تحديد مواقع محطات الشحن الكهربائي وتسهيل الحجز والدفع للموقف والشحن. باستخدام منهجيات الأجايل، يهدف المشروع إلى تحسين التنقل الحضري، وتعزيز الاستدامة البيئية من خلال تشجيع استخدام السيارات الكهربائية، والمساهمة في تقليل الازدحام الحضري. من خلال التكنولوجيا المبتكرة، يضع المشروع معيارًا جديدًا لحلول مواقف السيارات الحضرية الذكية والصديقة للسيارات الكهربائية، مبسطًا تجربة البحث عن مواقف في المشهد الحضري الحديث |
|  |

**Table of Contents**

List of Figures 9

List of Tables 11

Chapter 1 : Introduction 12

1. Background of problem statement 12

2. Description of the Problem 12

3. Description of the Suggested Solution 13

4. Literature Review (related work) 14

5. Technology and tools to be used 14

6. Organization of the Report 17

Chapter 2 : Project Plan 20

1.Project Objectives 20

2. Project Scope 20

3.Software Process Model 21

4. Project Schedule 22

5. Team Structure and Roles 22

Chapter 3 : Requirements and analysis 23

1. Functional Requirements 23

2. Non-functional Requirements 24

Chapter 4 : Architecture and Design 26

1. Architecture 26

2.Use Case Diagram 29

3. Activity Diagrams 44

4. Sequence Diagrams 55

5. Class Diagram 65

6. Entity Relationship Diagram 66

Chapter 5: Conclusion & Future Work 67

1.Summary of Achievements 67

2.Evaluation of Project Goals 67

3.Limitations and Challenges 68

4.Recommendations for Future Work 68

5.Concluding Remarks 68

List of Figures

Figure 2.1.1 Methodology 21

Figure 2.4.1 Project Schedule 22

Figure 2.5.1 Team Structure and roles 22

Figure 4.1.1 Architecture Diagram 26

Figure 4.2.1 Use case diagram 29

Figure 4.3.1 Register Activity Diagram 44

Figure 4.3.2 Login Activity Diagram 44

Figure 4.3.3 Subscribe Activity Diagram 45

Figure 4.3.4 Reservation Activity Diagram 46

Figure 4.3.5 Manage User Profile Activity Diagram 47

Figure 4.3.6 Forget Password Activity Diagram 48

Figure 4.3.7 Navigate To Spot Activity Diagram 49

Figure 4.3.8 Create Spot Activity Diagram 50

Figure 4.3.9 Delete Spot Activity Diagram 51

Figure 4.3.10 Modify Spot Activity Diagram 52

Figure 4.3.11 Check-In Activity Diagram 53

Figure 4.3.12 View Parking Spots Activity Diagram 54

Figure 4.4.1 Register Sequence Diagram 55

Figure 4.4.2 Login Sequence Diagram 56

Figure 4.4.3 Reset Password Sequence Diagram 57

Figure 4.4.4 Subscribe Sequence Diagram 58

Figure 4.4.5 Reservation Sequence Diagram 59

Figure 4.4.6 Create Spot Sequence Diagram 60

Figure 4.4.7 Modify & Delete Spot Sequence Diagram 61

Figure 4.4.8 Check-In at Reserved Parking Spot Sequence Diagram 62

Figure 4.4.8 Navigate to Reserved Spot Using Google Maps Sequence Diagram 63

Figure 4.4.9 View Parking Availability Sequence Diagram 64

Figure 4.5.1 Class Diagram 65

Figure 4.6.1 Entity Relationship Diagram 66

List of Tables

Table 4.2.1 Signup Use Case 30

Table 4.2.2 Login Use Case 31

Table 4.2.3 Forget Password Use Case 32

Table 4.2.4 OTP Use Case 33

Table 4.2.5 Subscribe Use Case 34

Table 4.2.7 Manage User Profile Use Case 35

Table 4.2.8 Check-In at Reserved Parking Spot Use Case 36

Table 4.2.5 Reserve Parking Spot Use Case 37

Table 4.2.6 Navigate to Reserved Spot Using Google Maps Use Case 38

Table 4.2.6 View Parking Availability Use Case 39

Table 4.2.9 Admin Authentication Use Case 40

Table 4.2.10 Add Parking Spots Use Case 41

Table 4.2.11 Modify Parking Spot Use Case 42

Table 4.2.12 Delete Parking Spots Parking Spot Use Case 43

Chapter 1 : Introduction

This chapter is about illustrating the description of the challenge of Solving the absence of a Urban Parking Solutions Mobile Application ”ParkEase” , related work to be done and the technology and tools that were going to use later so we can implement our website

1. Background of “problem statement”

The project emerges as a solution to ongoing challenges in urban parking design, especially in densely populated areas. With the ever-increasing number of cars, parking spaces have become increasingly scarce in cities worldwide, leading to traffic congestion and environmental issues Effective mobile applications' lack of resources to address these issues is a major compounding problem. This project aims to transform how parking is accessed and managed in cities through advanced mobile application technology.

2. Description of the Problem

Metropolitan areas worldwide are facing a shortage of accessible parking spots, leading to increased traffic jams, frustration among drivers, and environmental harm. Traditional parking systems often lack up-to-date information on available spots, resulting in inefficient utilization of current spaces and worsening the problem. Furthermore, the lack of integration with new technologies like electric vehicle (EV) charging infrastructure creates additional challenges for drivers and parking operators. This initiative aims to tackle these issues by maximizing the use of existing parking areas through real-time data, reducing traffic congestion, and decreasing emissions and pollution. It also incorporates EV charging stations to make it easier for drivers to access and promotes environmentally friendly urban development. Notably, drivers have the freedom to select their parking spot and are responsible for parking their vehicles, enhancing their overall experience. As a result, traffic flow has improved, environmental effects have been lessened, and the implementation of advanced mobile technologies has enhanced the overall efficiency and user experience in parking management.

3.Description of the Suggested Solution

The suggested remedy is an all-encompassing mobile app created to simplify the process of parking. By using up-to-the-minute information, the application enables individuals to find, book, and pay for parking spaces with just a few clicks on their mobile devices. For electric vehicle (EV) drivers, the app provides the extra feature of filtering spaces that have charging stations, meeting the increasing demand in the market. It links with Google Maps to assist in navigating to the selected parking space, offering a smooth and uninterrupted experience from booking to arrival.

Key features of "**ParkEase**" include:

* User Authentication : Secure and swift login process.
* Real-Time Parking Availability: Updated information on parking spots.
* In-App Reservation and Subscription: Users can reserve spots and subscribe for premium features.
* EV Charging Spot Identification: Specifically cater to EV owners.
* Integrated Navigation: Leverage Google Maps for user convenience.
* Profile Management: Users can manage their personal details and preferences.
* Administrative Control: Admins can add, modify, or delete parking spot details, ensuring accurate and current information.

4. Literature Review (related work)

Smart parking systems have emerged as a pivotal solution to mitigate the issues related to urban parking, traffic congestion, and environmental impact. Various studies and implementations have demonstrated the effectiveness of technology-driven solutions in optimizing parking space utilization and reducing the time spent searching for parking spots. This literature review explores several key studies and projects that have contributed to the development and advancement of smart parking systems.

1. Technology-Driven Smart Parking Solutions

Khan Aftab et al. (2020) present a comprehensive case study on reducing parking space search time and environmental impacts through a technology-driven smart parking solution. Their study highlights the increasing urbanization and the resultant demand for efficient transportation and parking solutions. The ParkUs solution proposed in the study leverages crowdsourced data and advanced infrastructure management to provide significant benefits for citizens, businesses, and cities (Aftab et al., 2020). This approach is indicative of the broader trend towards integrating IoT and AI in urban planning to enhance the efficiency of public services.

2. Sensor-Based Parking Systems

The use of sensors in smart parking systems has been extensively researched. An early study by Abbas and Gabriel (1998) introduced a differential global positioning system (DGPS) combined with a fuzzy logic Kalman filter for accurate positioning of vehicles in parking facilities. This innovative use of sensor fusion techniques set the stage for future developments in the field, emphasizing the importance of precise location tracking in parking management systems (Abbas & Gabriel, 1998).

3. Optimization of Traffic and Parking Management

Further advancements in traffic and parking optimization have been explored by Cheng et al. (2010), who focused on optimizing freeway traffic sensor locations through clustering GPS-derived speed patterns. Their research aimed to enhance the accuracy and efficiency of traffic monitoring systems, which are crucial for real-time parking management and the reduction of traffic congestion in urban areas (Cheng et al., 2010).

4. Crowdsourced Smart Parking Solutions

The promise of crowdsourced data in smart parking solutions has been demonstrated in several studies. Crowdsourcing provides a cost-effective method to collect real-time data on parking space availability, which can be integrated with smart parking applications to guide drivers to available spots. This approach not only reduces the time spent searching for parking but also helps in reducing fuel consumption and emissions.

5. Integration of AI in Parking Management

The integration of artificial intelligence in parking management has opened new avenues for developing intelligent and adaptive systems. AI algorithms can analyze vast amounts of data to predict parking space availability and optimize the allocation of parking resources. These systems can learn from patterns and trends, continuously improving their performance over time.

5. Technology and tools to be used

The development and deployment of the urban parking solutions mobile application, especially designed to accommodate electric vehicles (EVs), necessitates the use of modern technologies and tools. This comprehensive approach ensures the application is robust, user-friendly, and scalable, addressing the diverse needs of urban drivers and EV owners. The selection of technologies and tools for this project includes:

**Frontend (Client Application)**

* Flutter: Remains the best choice for a cross-platform mobile application, enabling a unified codebase for both iOS and Android.
* Figma: For UI/UX design, essential for crafting an intuitive and appealing user interface.

**Backend Services**

* Node.js: Provides the runtime environment for your server-side logic, known for its efficiency and scalability.
* Express.js: A web application framework for Node.js that simplifies API development with its robust set of features.
* MongoDB Atlas: A cloud-based database service that offers scalability and flexibility, perfect for storing user data, parking information, and reservations.
* Mongoose: Enhances MongoDB usage with schema definitions, data validation, and query building.

**Payment Integration**

* 2CheckOut: For processing global payments, supporting multiple payment methods and currencies, integrated into your application to handle transactions securely.

**Deployment and Hosting**

* Railway: To deploy and manage your application's backend services, providing an easy path to scaling and monitoring.
* MongoDB Atlas: For the database, leveraging its cloud infrastructure for high availability and performance.

**DevOps and Version Control**

* GitHub: For source code management, collaboration, and version control.

**Authentication**

* JWT (JSON Web Tokens): For authentication and session management, ensuring secure and scalable user authentication within your application.
* Rapid SMS API: For sending OTPs to users' phone numbers, enhancing security by verifying user identity during sign-up and login processes.

**Development and Testing Tools**

* Postman: For API testing, ensuring that your backend services are correctly implemented before they go live.
* Jest: For writing unit and integration tests, particularly for your Node.js backend, ensuring code reliability and functionality.

**Monitoring and Analytics**

* MongoDB Atlas Monitoring: For insights into database performance, helping identify and resolve potential issues quickly.
* Railway Metrics and Logs: For real-time monitoring of your application’s health and performance.

**Other Tools**

* Google Maps API: Essential for integrating map and location services, crucial for navigating to parking spots.
* SSL/TLS Certificates: To secure communications between the client and server, protecting user data during transmission.

6. Organization of the Report

* **Chapter 1: Introduction**

This chapter sets the stage for the report, presenting the problem statement around the lack of a centralized academic resource platform. It provides a thorough description of the problem, outlining the challenges students face in accessing up-to-date and relevant academic materials. The suggested solution, the “ **ParkEase”**, is introduced as a digital platform tailored to the needs of university students and faculty. This chapter also reviews related literature and details the technologies and tools utilized in the development of “**ParkEase”**.

* **Chapter 2: Project Plan**

The project plan chapter outlines the objectives, scope, and the software process model of the **ParkEase** project. It details the strategic plan for achieving the project's goals, defining the limits of the project scope, and describing the Waterfall methodology adopted for structured development. The project schedule, including Gantt charts, and the roles and responsibilities within the team are also presented, providing a comprehensive roadmap for the project's execution.

* **Chapter 3: Requirements and Analysis**

Chapter 3 delves into the functional and non-functional requirements that frame the development of **ParkEase**. It lists the essential features and specifications that the system must meet, such as user authentication, resource cataloging, and search functionality. The chapter also defines performance, security, and usability requirements, ensuring the system's reliability and efficiency.

* **Chapter 4: Architecture and Design**

The architecture and design chapter describes the structural blueprint of the **ParkEase**, highlighting the layered approach consisting of the presentation, application, and data layers. It includes detailed design components like use case diagrams, sequence diagrams, activity diagrams, class diagrams, and the entity-relationship diagram, illustrating the system's design considerations and database schema.

* **Chapter 5: System Implementation**

In this chapter, the implementation details are discussed, covering the development environment, database implementation, user interface construction, and functionality implementation. The integration of external services/APIs, security measures, and performance optimization techniques are also elaborated upon, showcasing the practical application of the proposed design and architecture.

* **Chapter 6: System Testing and Installation**

Chapter 6 addresses the testing and installation processes of the ParkEase. It describes the various testing strategies employed, such as unit, integration, system, and user acceptance testing, as well as security and performance testing. The installation process is detailed to ensure the correct deployment of the system, and a thorough validation of requirements is conducted to confirm that the system fulfills the established criteria.

* **Chapter 7: Conclusion and Future Work**

The final chapter wraps up the report with a summary of the project's achievements and an evaluation of how the project goals were met. It acknowledges the limitations and challenges encountered during the project and proposes recommendations for future enhancements. The concluding remarks reflect on the project's contribution to academic resource

Chapter 2 : Project Plan

1.Project Objectives

The purpose of the mission is to create "ParkEase" a cellular app aimed at making parking in towns a breeze. It plans to try this through imparting up to date information on parking spots permitting users to reserve and pay for these spots without delay from their phones which include people with electric car (EV) chargers. The app ambitions to reduce down the time drivers spend looking for parking inspire the usage of EVs and provide specific subscription levels for added advantages. All the even as it is focused on being smooth to scale up stable from digital threats and reliable regardless of how many human beings are using it aiming to make urban driving smoother and more eco-friendly.

2. Project Scope

The scope encompasses the development of a cross-platform application accessible on both iOS and Android, featuring:

* Real-time parking and EV charging spot availability.
* A subscription model.
* Secure in-app payment and reservation system.
* User-friendly design and navigation facilitated by Figma and Google Maps API.

Exclusions include global coverage at launch, direct involvement in parking enforcement, and the development of physical parking infrastructure

3.Software Process Model

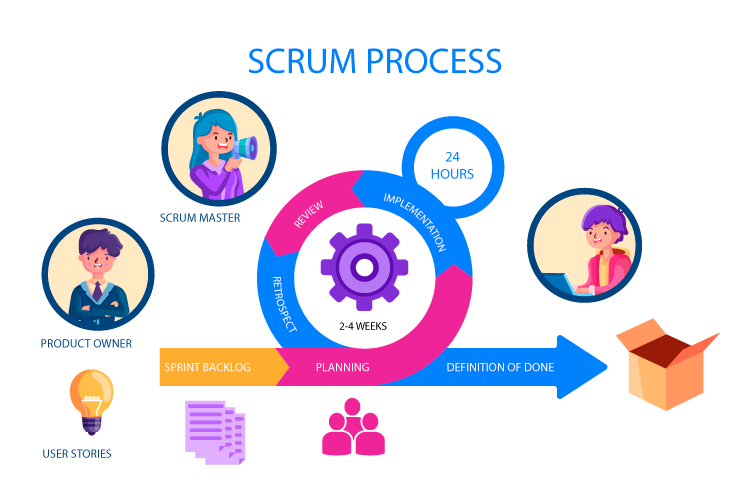
The project will adopt the Agile development methodology, specifically utilizing the Scrum framework to allow for flexibility, rapid iteration, and continuous feedback throughout the development process. This approach supports the dynamic nature of the project and enables the team to adapt to user needs and technological advancements efficiently.

Figure 2.1.1 Methodology

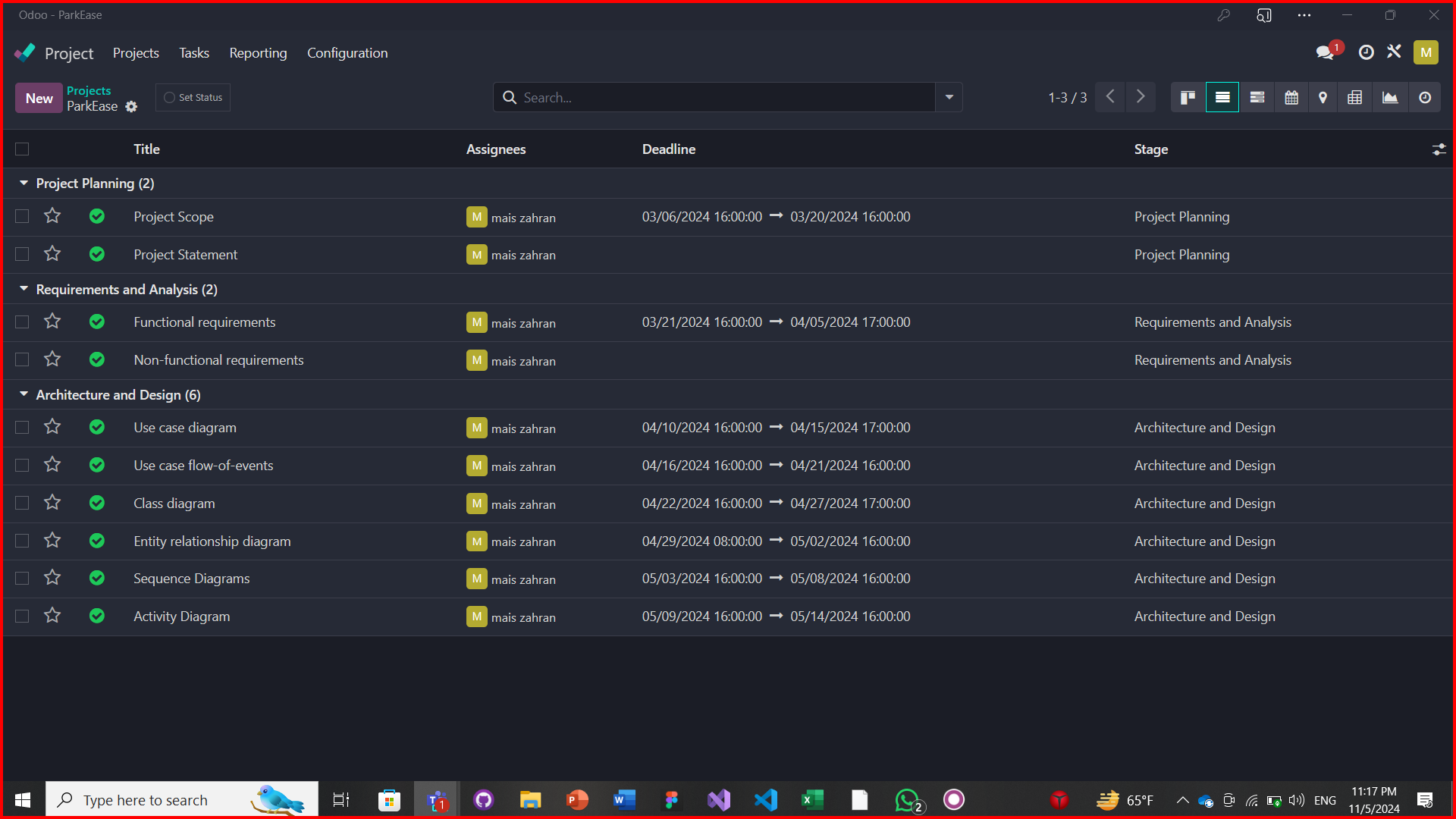
4. Project Schedule

Figure 2.4.1 Project Schedule

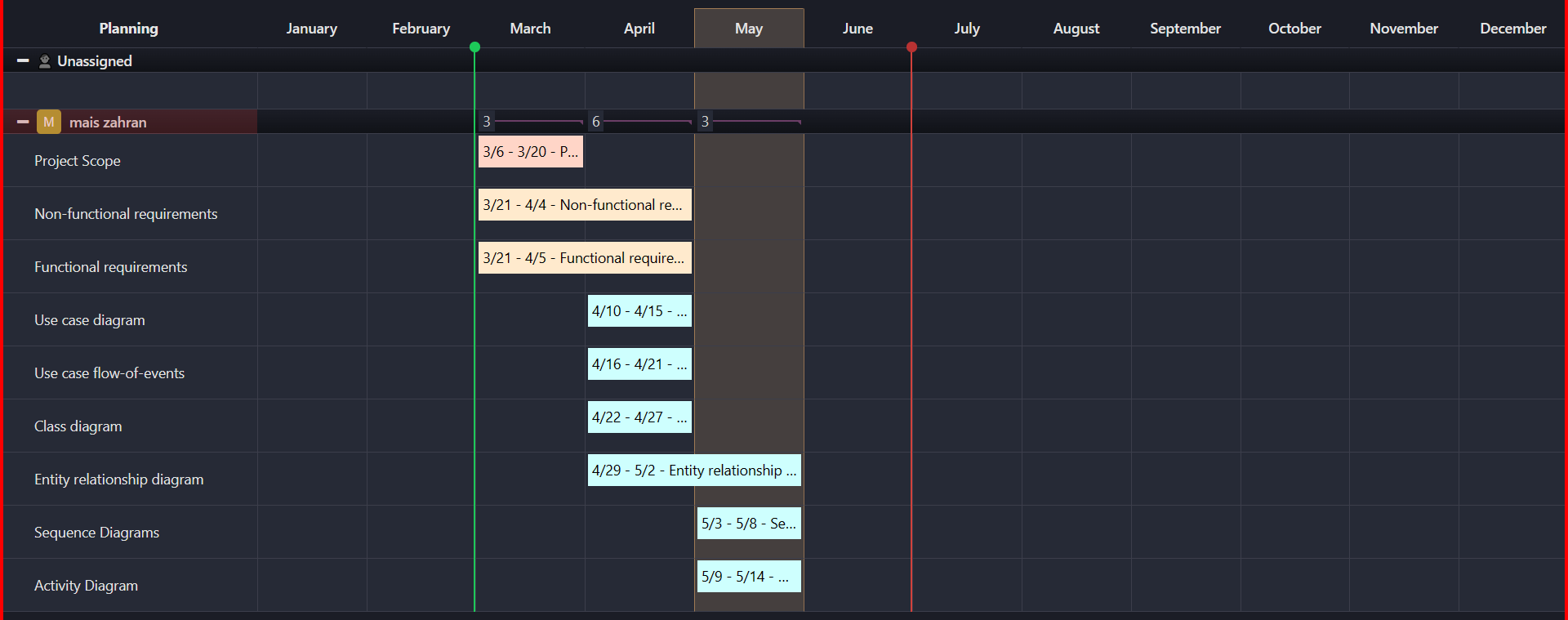
5. Team Structure and Roles

Figure 2.5.1 Team Structure and roles

Chapter 3 : Requirements and analysis

1. Functional Requirements

Functional requirements specify the essential behaviors and functionalities the application must support to meet user needs and project objectives:

**User Authentication:**

* + Users must be able to log in and sign up using there number or username.
  + Authentication should be secure and comply with OAuth 2.0 standards.

**Parking Spot Reservation:**

* + Users must be able to view the availability of parking spots in real-time.
  + The app should allow users to reserve a parking spot.
  + Reservation includes spots with EV chargers where applicable.

**Subscription Service:**

* + Users should be able to subscribe to the app for reservation features.
  + Payment processing for subscriptions must be secure and support major credit/debit cards.

**Navigation Integration:**

* + Once a parking spot is reserved, the app must provide an option to navigate to the spot using Google Maps.

**EV Charging Spot Feature:**

* + The app must allow filtering of parking spots to show only those with EV charging facilities.

**Profile Management:**

* + Users must be able to manage personal details and payment methods.

**Admin Management:**

* + Admins should be able to add, modify, or delete parking spot information.

2. Non-functional Requirements

Non-functional requirements describe the standards the application must adhere to regarding performance, security, usability, and scalability:

**Performance:**

* + The app should load parking availability within 3 seconds.
  + Reservation transactions should not take more than 5 seconds to process.

**Usability:**

* + The app interface must be user-friendly and accessible to users with no technical background.
  + It should be intuitive to navigate within the app.

**Reliability:**

* + The app should have an uptime of 99.9%.
  + Real-time data should have a 98% accuracy rate.

**Scalability:**

* + The system should be scalable to accommodate a growing number of users and parking spots.
  + It must handle at least 1000 concurrent users.

**Security:**

* + Data encryption for personal and payment information must adhere to industry standards.
  + The app must comply with GDPR and other relevant privacy regulations.

**Compatibility:**

* + The app must be compatible with both iOS and Android platforms.
  + It should support the latest and two previous versions of the operating systems.

**Maintainability:**

* + The app should be easy to maintain and update without significant downtime.

**Localization:**

* + The app should support multiple languages, starting with English and expanding as per the user base.

**Chapter 4 : Architecture and Design**

1. Architecture

**Overview**

This section delineates the layered architecture of the **ParkEase** , ensuring a seamless user experience and robust system functionality. The architecture is composed of three primary layers—Presentation Layer, Application Layer (inclusive of Business Logic), and Data Layer.

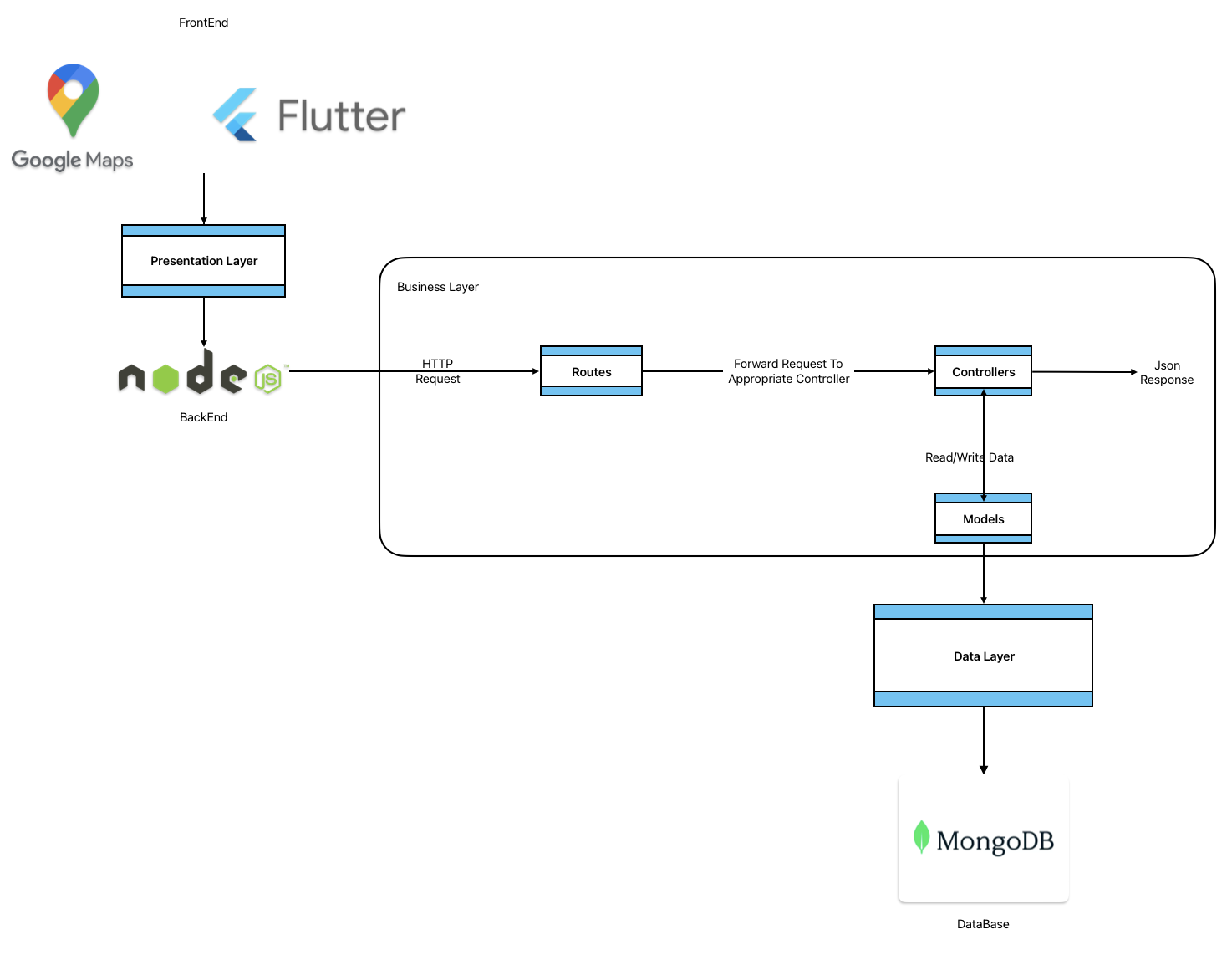
**Architecture Diagram**

Figure 4.1.1 Architecture Diagram

High-Level Architecture of ParkEase

The application's architecture is designed to be modular and scalable, comprising several key components:

**1. Presentation Layer:**

* Technology: Flutter for cross-platform mobile development; Figma for UI/UX design.
* Features: Native performance on iOS and Android, intuitive design for ease of use, secure data handling and transmission.

**2. Application Layer:**

* Core Technologies: Node.js with Express.js for backend services; JWT for authentication; Rapid SMS API for OTP verification.
* Design Principles: Microservices architecture for modular, scalable development and deployment; API Gateway for centralized request handling and security enhancements.
* Integration: Google Maps API for real-time navigation and mapping; 2CheckOut for secure payment processing.

**3. Data Layer:**

* Database: MongoDB Atlas for a flexible, scalable NoSQL database solution; Mongoose for data modeling and validation.
* Security: Encryption of sensitive data, secure access controls, regular data backups.

**4. Deployment & Monitoring:**

* Deployment: Railway for backend hosting with CI/CD pipelines via GitHub Actions for automated testing and deployment.
* Monitoring: Railway Metrics and Logs for real-time application performance monitoring; integration with third-party tools for in-depth analysis.

**5. Security & Compliance:**

* Approach: Implement SSL/TLS for secure communication; adhere to PCI DSS standards for payment processing; GDPR compliance for data protection.
* Authentication: Stateless authentication using JWT for a secure and efficient user login process; Rapid SMS API integration for OTP-based phone verification.

2.Use Case Diagram

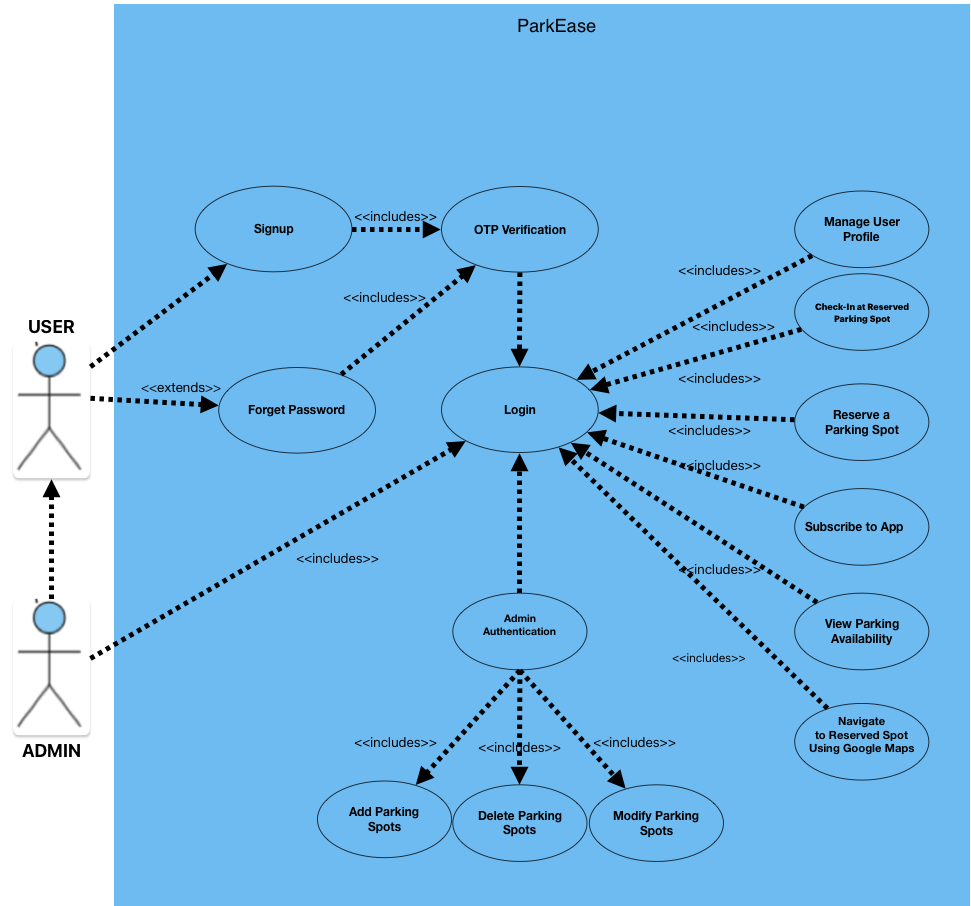


Figure 4.2.1 Use case diagram

Table 4.2.1 Signup Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Signup** |
| Goal | Register and create a new user account in the app |
| Preconditions | User has the app installed and does not have an account |
| Actors | End User |
| Main Success Scenario | 1. User opens the app and selects “Signup". 2. User enters required information (e.g., email, phone number). 3. User sets a password. 4. User receives OTP via SMS (Rapid SMS API). 5. User enters the received OTP. 6. User's information is verified and account is created. 7. User is taken to the login screen or directly logged in. |
| Extensions | 1a. User tries to register with an email that is already in use.  1b. User is informed and asked to try a different email or log in.  2a. User enters an invalid phone number.  2b. User is prompted to enter a valid phone number.  5a. User enters an incorrect OTP.  5b. User is allowed to try again or resend OTP. |
| Postconditions | User's account is created and user is logged in or ready to log in. |
| Quality Requirements | The signup process must be completed within 5 minutes to prevent OTP expiry. |

Table 4.2.2 Login Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Login** |
| Goal | Authenticate and log into the app using an account |
| Preconditions | User has the app installed and has an existing account |
| Actors | End User |
| Main Success Scenario | 1. User opens the app and selects "Login". 2. User enters their credentials (e.g., email and password). 3. User's credentials are authenticated. 4. User is logged into the app. |
| Extensions | 2a. User enters incorrect credentials.  2b. User is informed of the error and can try to enter credentials again.  2c. User selects "Forgot Password" to initiate password recovery.  3a. Backend service is unavailable.  3b. User is informed and asked to try again later. |
| Postconditions | User is logged into their account and directed to the app's main interface. |

Table 4.2.3 Forget Password Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Forget Password** |
| Goal | Reset the user's forgotten password through OTP verification |
| Actors | End User |
| Main Success Scenario | 1. User selects "Forget Password" on the login screen. 2. User enters their registered phone number. 3. System validates the phone number and sends an OTP via SMS. 4. User enters the received OTP. 5. System verifies the OTP. 6. User is prompted to set a new password. 7. User sets a new password and is taken to the login screen or directly logged in. |
| Preconditions | User has an existing account with a verified phone number. |
| Postconditions | User's password is reset, and the user is logged in or ready to log in. |

Table 4.2.4 OTP Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Send OTP** |
| Goal | Send an OTP to verify the user's phone number |
| Actors | System |
| Main Success Scenario | 1. System triggers the OTP process in response to a user action (signup or password reset). 2. System generates a unique OTP. 3. System sends the OTP to the user's provided phone number via SMS (SMS Gateway API). 4. User receives the OTP. 5. System waits for user to enter the OTP for verification. |
| Preconditions | User has initiated a signup or forget password request. |
| Postconditions | OTP is sent to the user's phone number, awaiting verification. |

Table 4.2.5 Subscribe Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Subscribe to App** |
| Goal | Subscribe to the app's premium services via online payment. |
| Preconditions | User is logged in and has valid payment information registered or entered during the process. |
| Actors | End User |
| Trigger | User selects the option to subscribe to premium services. |
| Main Success Scenario | 1. User navigates to the subscription section in the app. 2. User reviews the subscription options and benefits. 3. User selects a subscription plan. 4. User is prompted to confirm payment method. 5. User confirms and authorizes payment. 6. Payment is processed securely via 2CheckOut. 7. User receives confirmation of successful subscription. |
| Extensions | 4a. Payment method is invalid or transaction fails.<br>4b. User is notified and asked to provide a different payment method.<br>4c. User selects to add a new payment method and proceeds with subscription.<br>5a. User decides to cancel the process.<br>5b. User exits the subscription process and no changes are made. |
| Postconditions | User has access to premium features and services based on the chosen subscription plan. |

Table 4.2.7 Manage User Profile Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Manage User Profile** |
| Goal | To allow the user to update or modify their profile details. |
| Preconditions | User is logged into the app. |
| Actors | End User |
| Trigger | User selects the "Profile" or "Settings" option. |
| Main Success Scenario | 1. User navigates to profile settings. 2. User updates desired profile details (e.g., name, contact info). 3. User submits the changes. 4. System validates and saves the new details. 5. User receives confirmation of update. |
| Extensions | 3a. User enters invalid data.  3b. System displays an error message and requests valid information.  4a. System is unable to save changes (e.g., server error).  4b. User is informed and asked to try again later. |
| Postconditions | User’s profile is updated with new details. |

Table 4.2.8 Check-In at Reserved Parking Spot Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Check-In at Reserved Parking Spot** |
| Goal | Confirm user's arrival at the reserved parking spot within the stipulated time. |
| Preconditions | User has made a parking reservation and is at the parking location. |
| Actors | End User (Driver) |
| Trigger | User arrives at the parking location and wants to check in. |
| Main Success Scenario | 1. User navigates to the reservation in the app. 2. User selects the option to check in. 3. App verifies the location of the user. 4. App confirms the check-in. 5. Parking spot status updates to "occupied". |
| Extensions | 3a. User attempts to check in from a different location.  3b. App denies check-in and informs user they need to be at the reserved location.  3c. User requests assistance or guidance to the correct parking spot. |
| Postconditions | The reservation is confirmed; the parking spot is occupied by the user. |

Table 4.2.5 Reserve Parking Spot Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Reserve a Parking Spot** |
| Goal | To allow the user to reserve an available parking spot. |
| Preconditions | User is logged in and has payment method on file. |
| Actors | End User |
| Main Success Scenario | 1. User selects desired parking location. 2. User selects the time and duration for reservation. 3. System calculates the cost and requests confirmation. 4. User confirms and completes payment. 5. System confirms reservation and provides details. |
| Extensions | 2a. Selected spot is no longer available.  2b. User is notified and asked to choose another spot.  3a. Payment method is declined.  3b. User is prompted to update payment information. |
| Postconditions | Parking spot is reserved for the user. |

Table 4.2.6 Navigate to Reserved Spot Using Google Maps Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Navigate to Reserved Spot Using Google Maps** |
| Goal | To guide the user to their reserved parking spot using navigation. |
| Preconditions | User has an active parking spot reservation. |
| Actors | End User |
| Main Success Scenario | 1. System retrieves the location of the reserved spot. 2. System launches Google Maps with the destination set. 3. Google Maps provides turn-by-turn navigation to the spot. |
| Extensions | 2a. Google Maps is unable to provide navigation (e.g., lack of GPS signal).  2b. User is notified and given alternative directions if possible. |
| Postconditions | User arrives at the reserved parking location. |

Table 4.2.6 View Parking Availability Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **View Parking Availability** |
| Goal | To display real-time availability of parking spots. |
| Preconditions | User is logged into the app. |
| Actors | End User |
| Main Success Scenario | 1. User selects location to view parking spots. 2. System displays available spots and relevant information (e.g., location, price). |
| Extensions | 1a. There are no available spots.  1b. System informs the user and may offer to notify when spots become available. |
| Postconditions | User is informed of parking spot availability. |

Table 4.2.9 Admin Authentication Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Admin Authentication** |
| Use Case ID | AD001 |
| Goal | Authenticate an admin and verify their role in the system. |
| Primary Actor | Admin |
| Preconditions | Admin has the app installed and has admin credentials. |
| Main Flow | 1. Admin selects the option to log in. 2. Admin enters their credentials. 3. System verifies credentials and checks the database for admin role. 4. Admin is granted access to admin functionalities. |
| Postconditions | Admin is logged into the system with access to admin-specific features. |
| Extensions | 1. Credentials are invalid; admin is prompted to retry. 2. Credentials do not grant admin access; user is informed they lack admin privileges. |

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Add Parking Spots** |
| Use Case ID | AD002 |
| Goal | Input new parking spots into the system |
| Primary Actor | Admin |
| Preconditions | Admin is authenticated and verified as an admin. |
| Main Flow | 1. Admin inputs details of a new parking spot 2. Saves the new parking spot in the system |
| Postconditions | New parking spot is available for users |
| Extensions | Details are invalid; admin corrects and resaves |

Table 4.2.10 Add Parking Spots Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Modify Parking Spots** |
| Use Case ID | AD003 |
| Goal | Update details of existing parking spots |
| Primary Actor | Admin |
| Preconditions | Admin is authenticated and verified as an admin. |
| Main Flow | 1. Admin selects a parking spot to modify 2. Updates details and saves changes |
| Postconditions | Parking spot details are updated |
| Extensions | Update fails; admin retries |

Table 4.2.11 Modify Parking Spot Use Case

|  |  |
| --- | --- |
| Element | Details |
| Use Case Name | **Delete Parking Spots** |
| Use Case ID | AD004 |
| Goal | Remove parking spots from the system |
| Primary Actor | Admin |
| Preconditions | Admin is authenticated and verified as an admin. |
| Main Flow | 1. Admin selects a parking spot to delete<br>2. Confirms deletion |
| Postconditions | Parking spot is removed from availability |
| Extensions | Deletion fails; admin retries |

Table 4.2.12 Delete Parking Spots Parking Spot Use Case

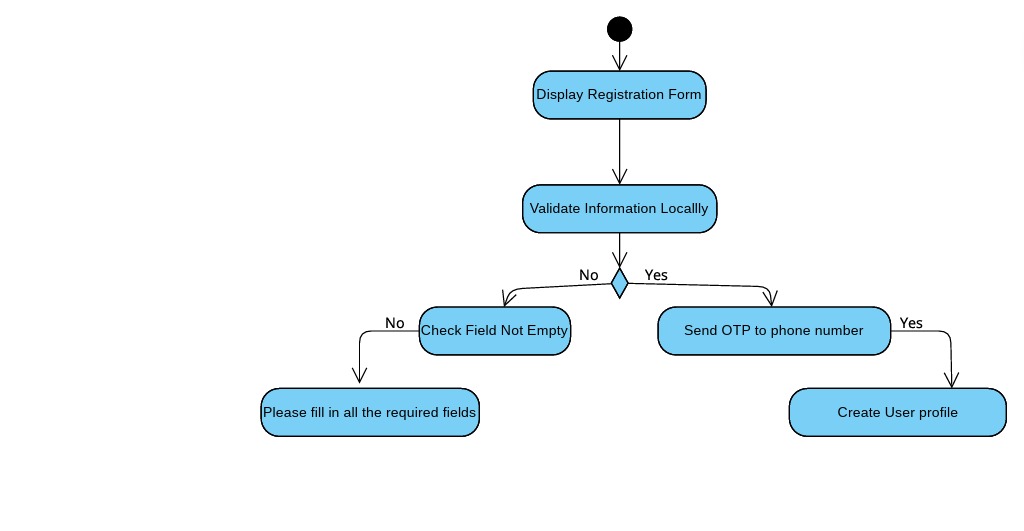
3. Activity Diagrams

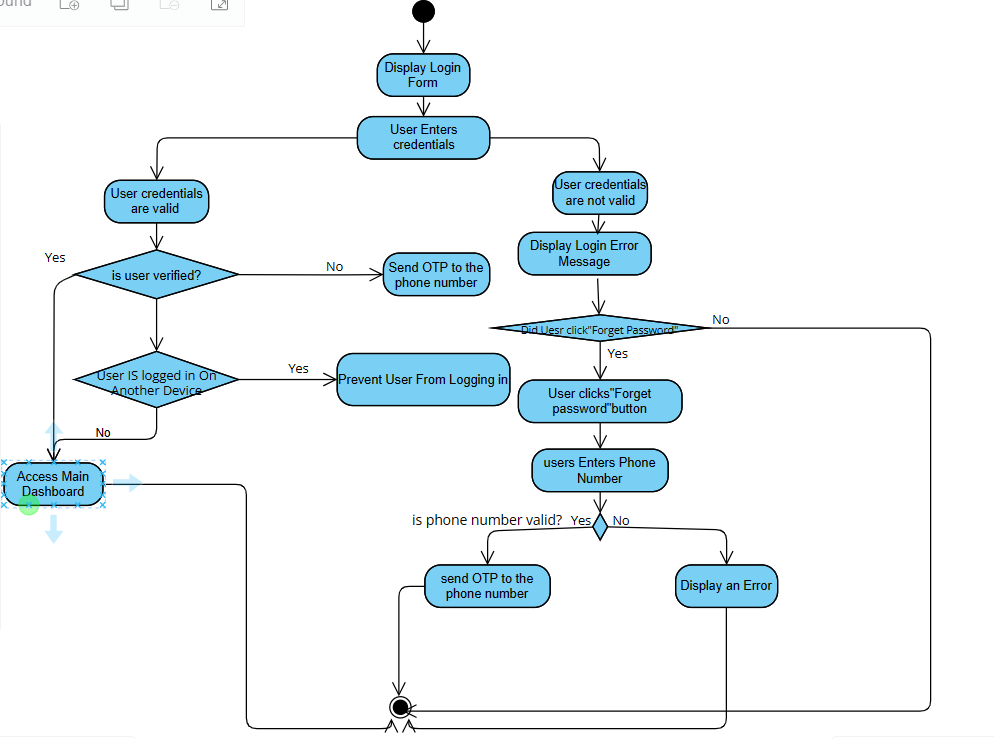
Figure 4.3.1 Register Activity Diagram

Figure 4.3.2 Login Activity Diagram

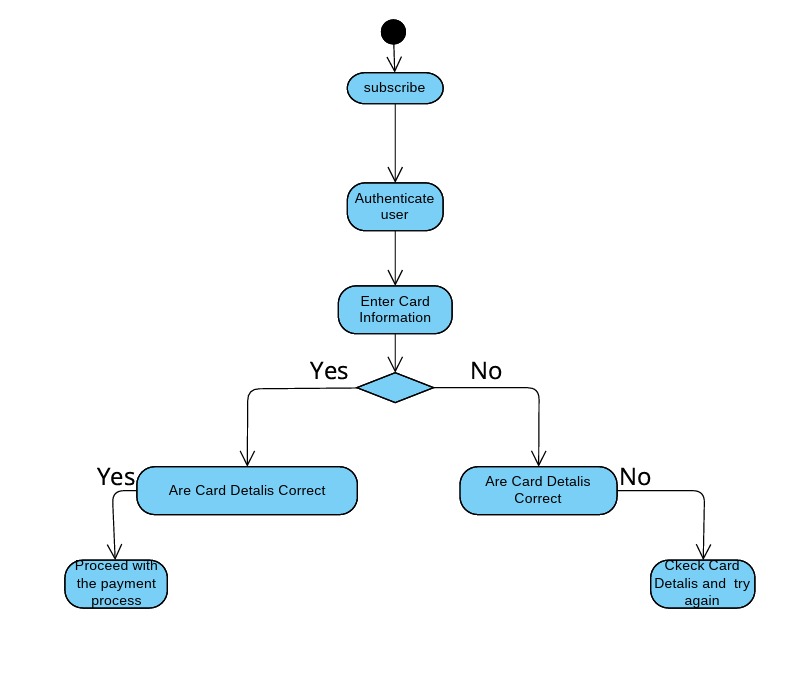


Figure 4.3.3 Subscribe Activity Diagram

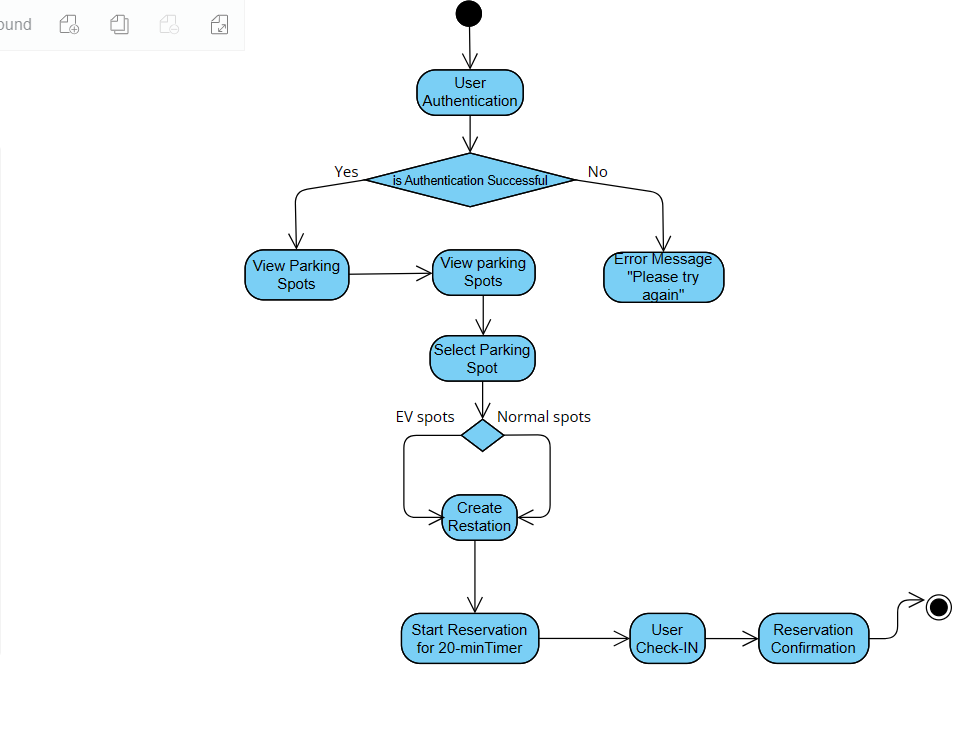


Figure 4.3.4 Reservation Activity Diagram

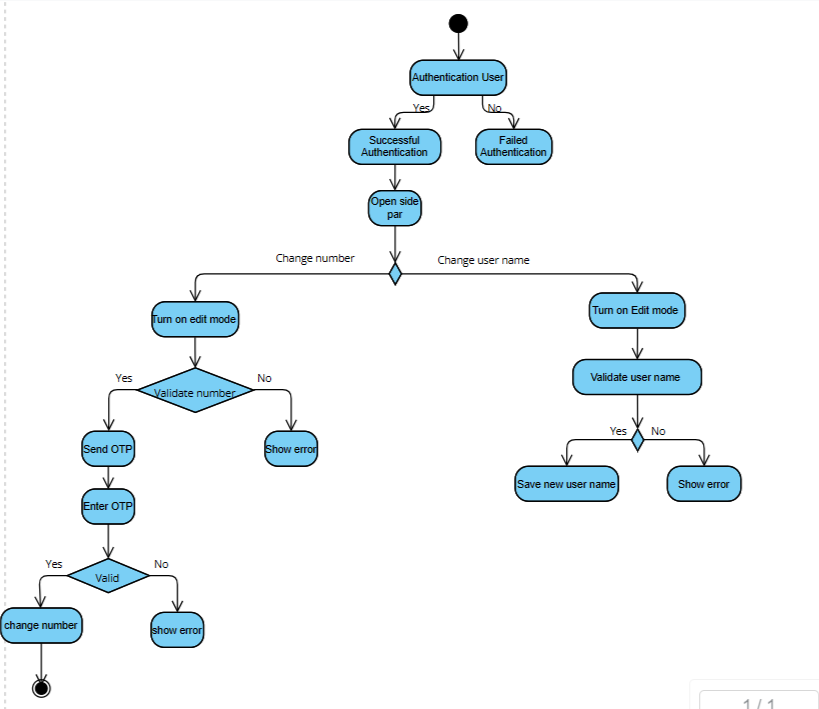
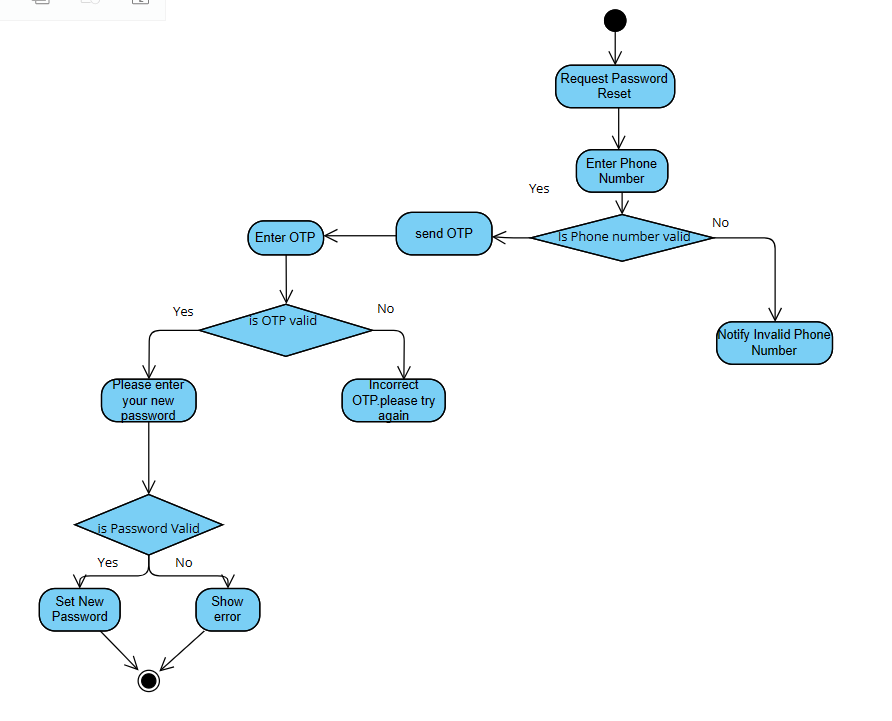


Figure 4.3.5 Manage User Profile Activity Diagram

Figure 4.3.6 Forget Password Activity Diagram

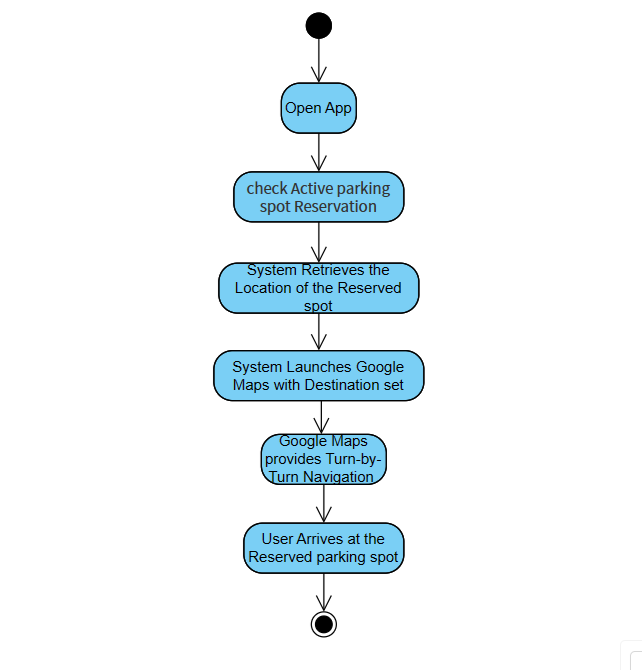


Figure 4.3.7 Navigate To Spot Activity Diagram

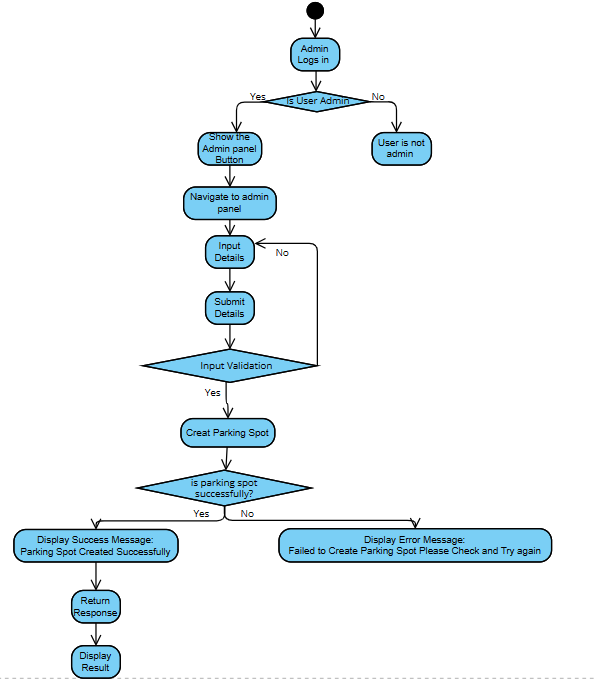


Figure 4.3.8 Create Spot Activity Diagram

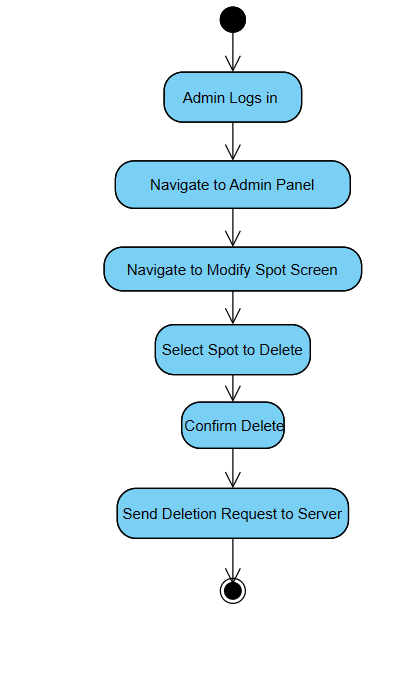


Figure 4.3.9 Delete Spot Activity Diagram

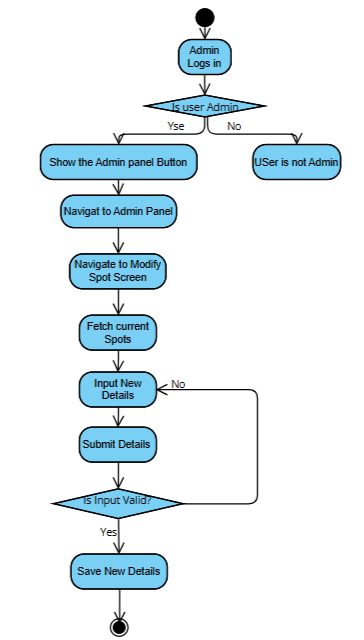
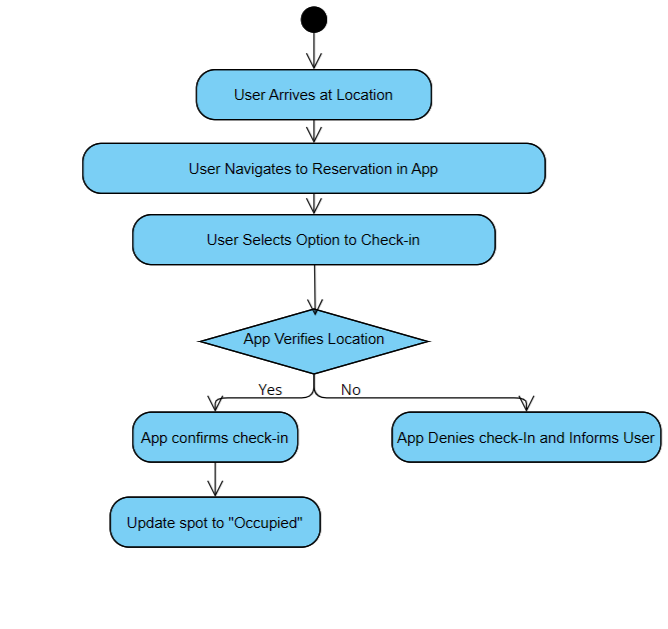


Figure 4.3.10 Modify Spot Activity Diagram

Figure 4.3.11 Check-In Activity Diagram



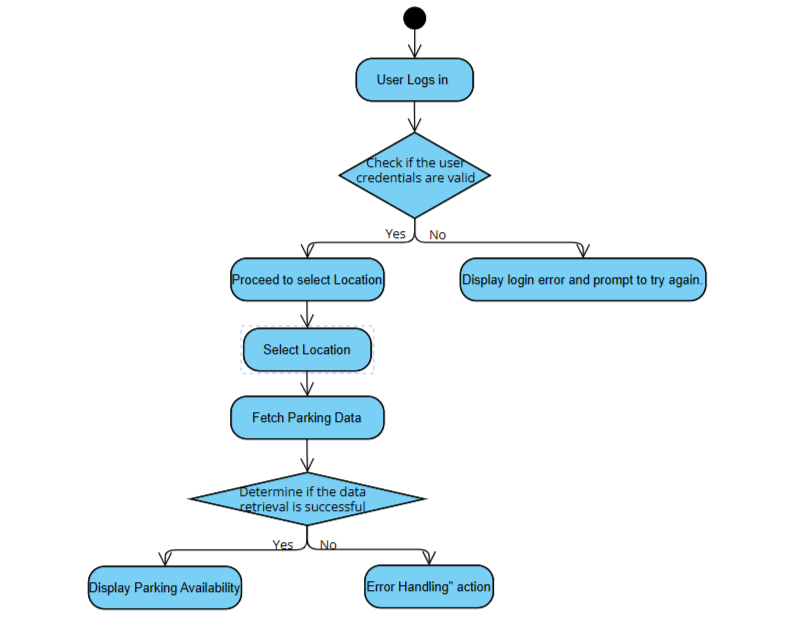


Figure 4.3.12 View Parking Spots Activity Diagram

4. Sequence Diagrams

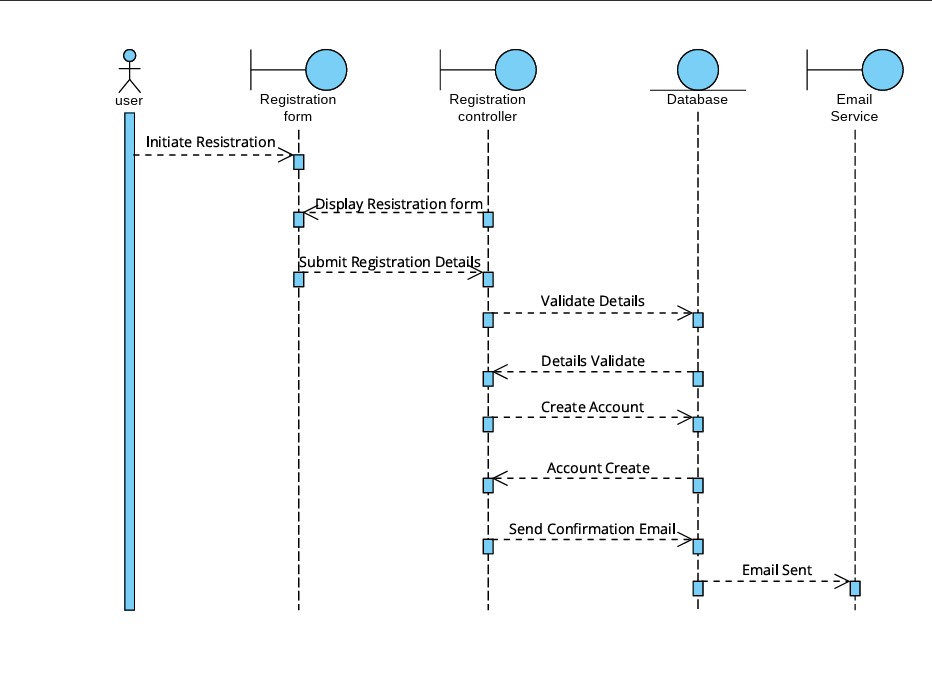


Figure 4.4.1 Register Sequence Diagram

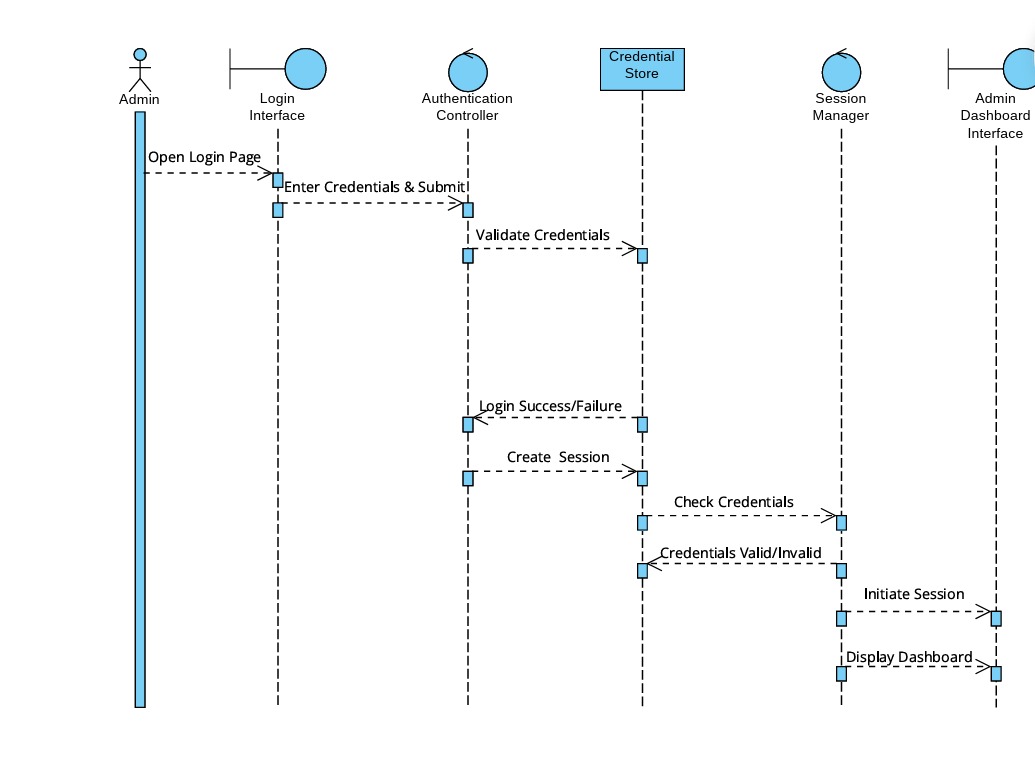


Figure 4.4.2 Login Sequence Diagram

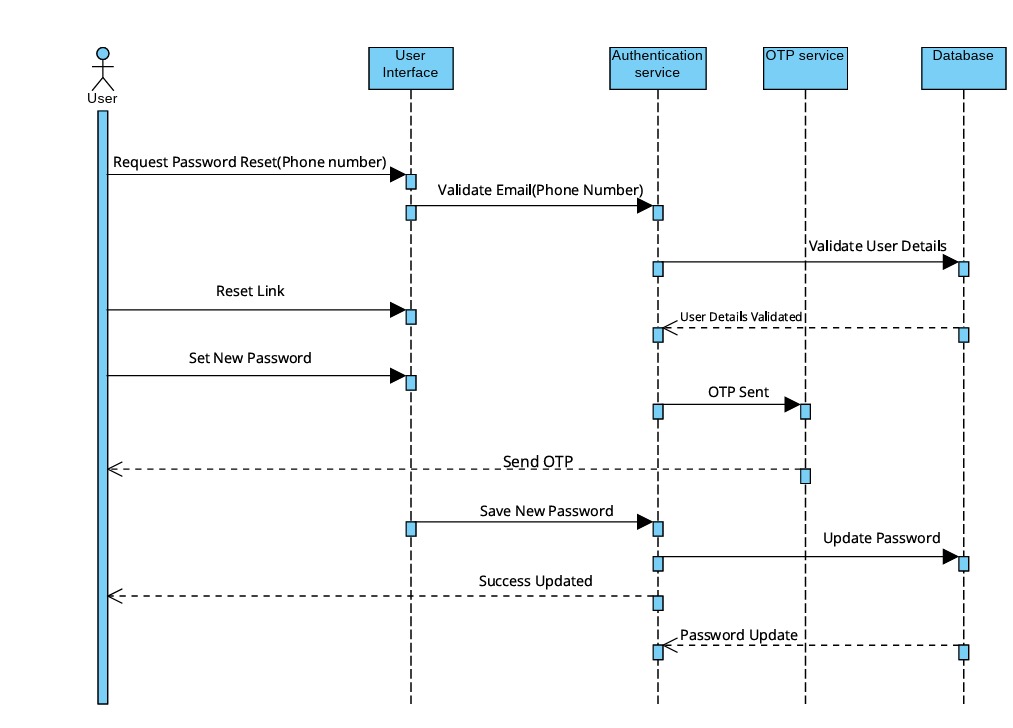


Figure 4.4.3 Reset Password Sequence Diagram

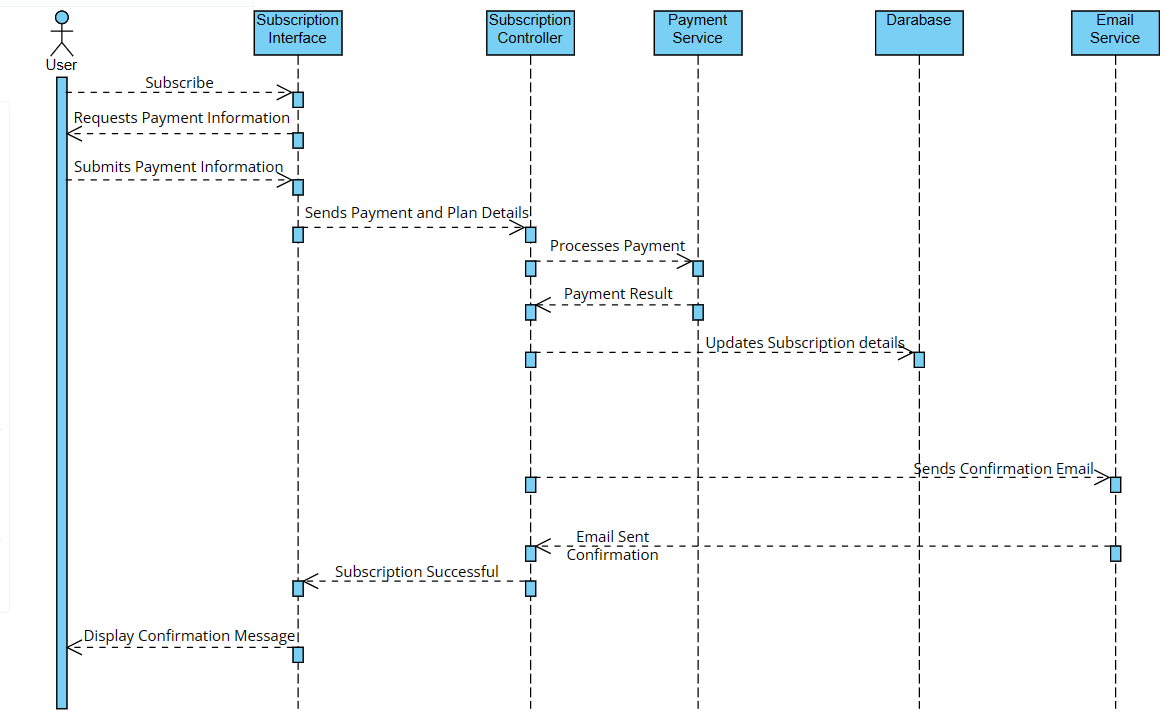


Figure 4.4.4 Subscribe Sequence Diagram

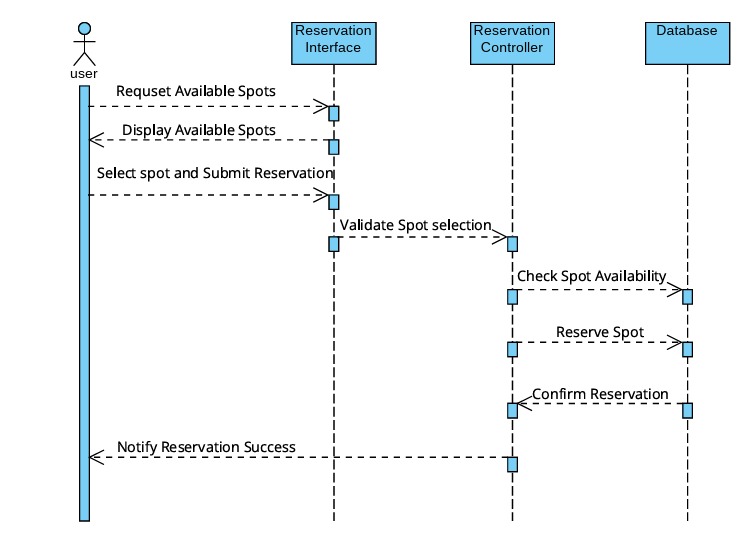


Figure 4.4.5 Reservation Sequence Diagram

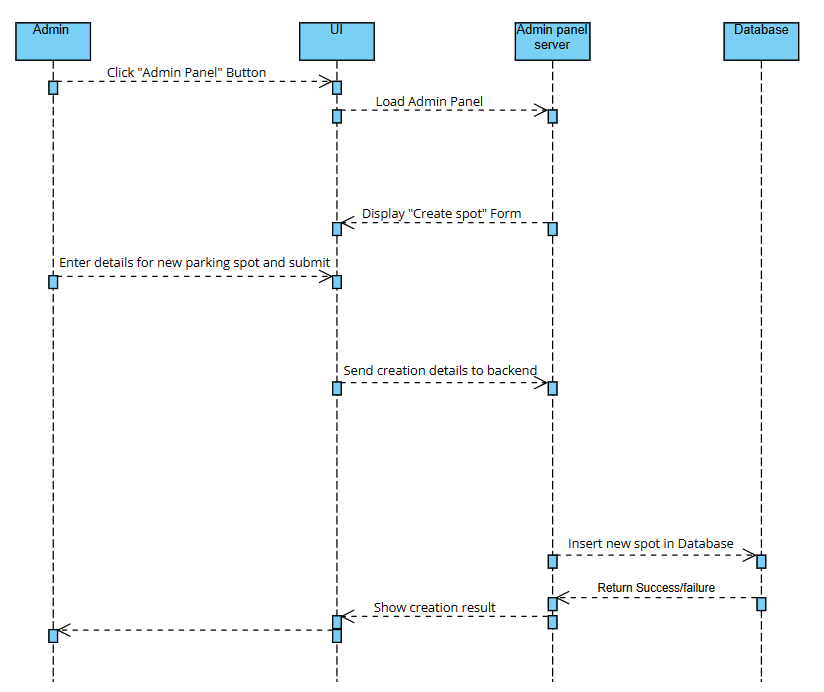
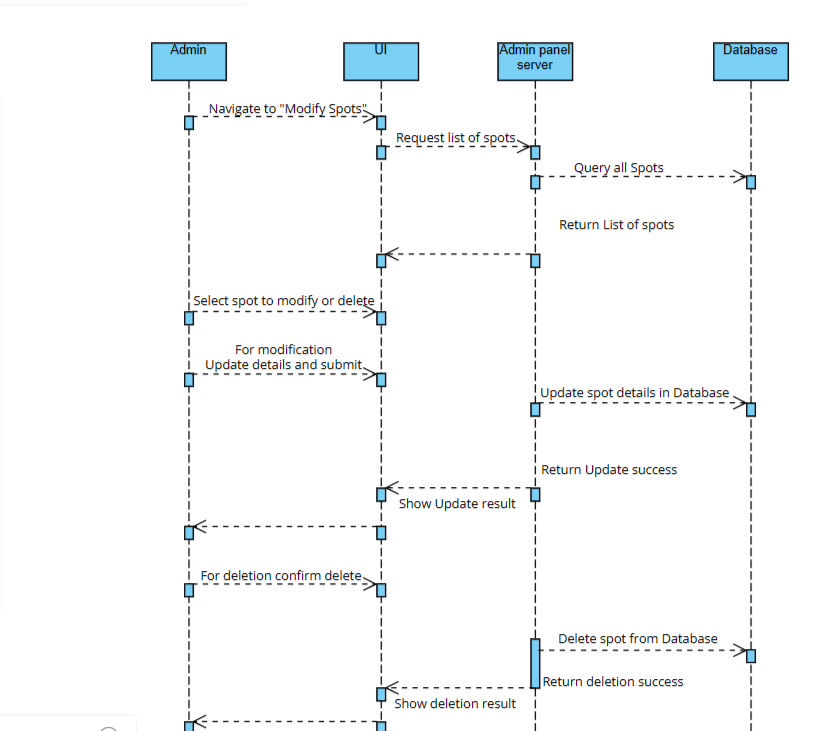


Figure 4.4.6 Create Spot Sequence Diagram

Figure 4.4.7 Modify & Delete Spot Sequence Diagram

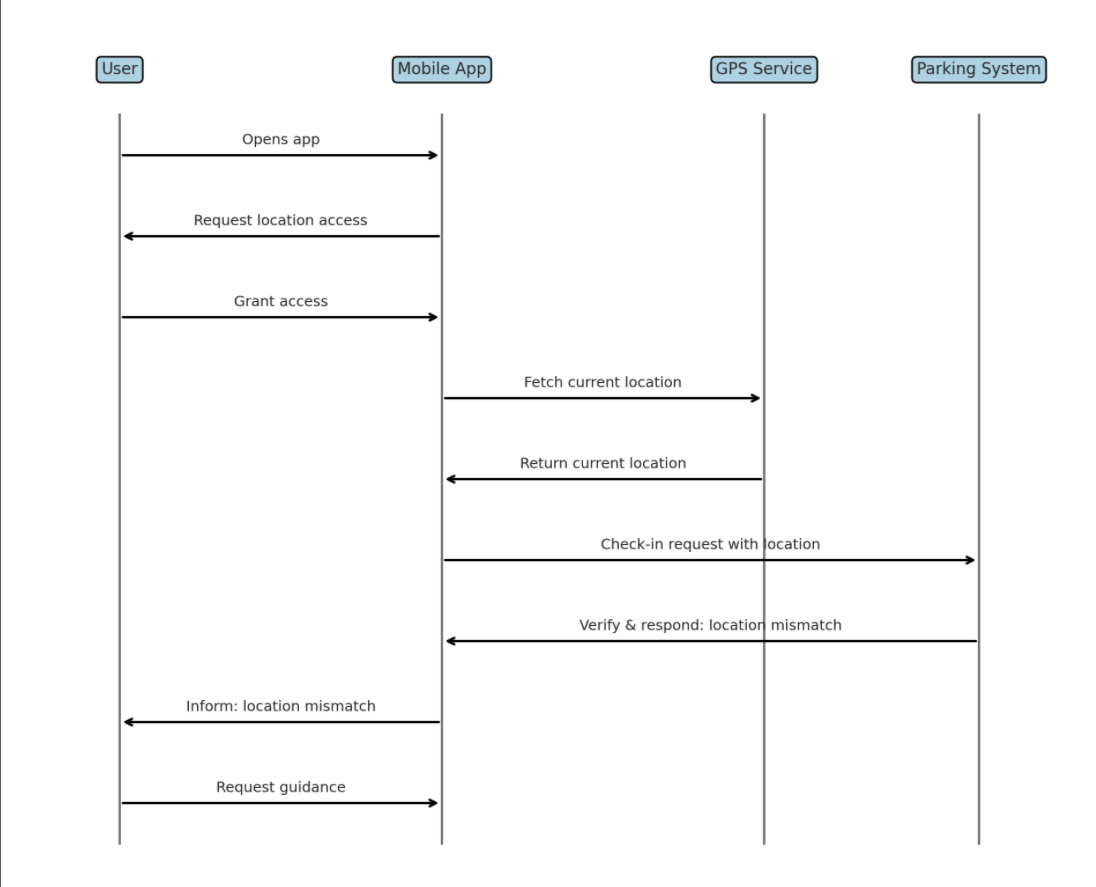


Figure 4.4.8 Check-In at Reserved Parking Spot Sequence Diagram

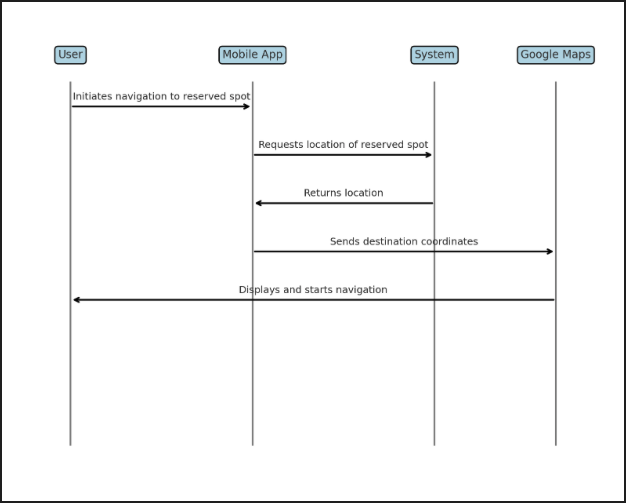


Figure 4.4.8 Navigate to Reserved Spot Using Google Maps Sequence Diagram

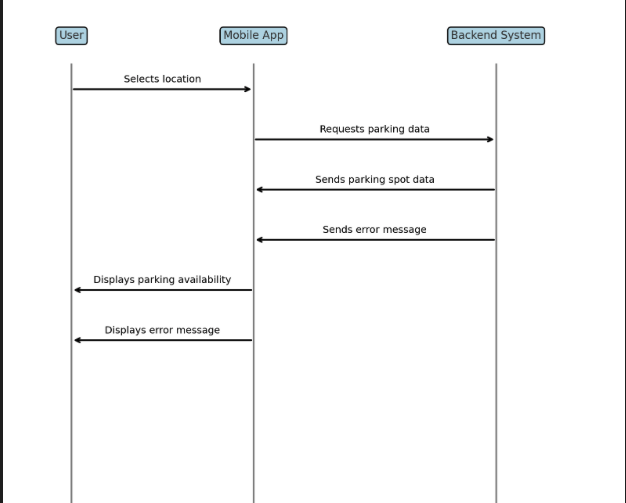


Figure 4.4.9 View Parking Availability Sequence Diagram

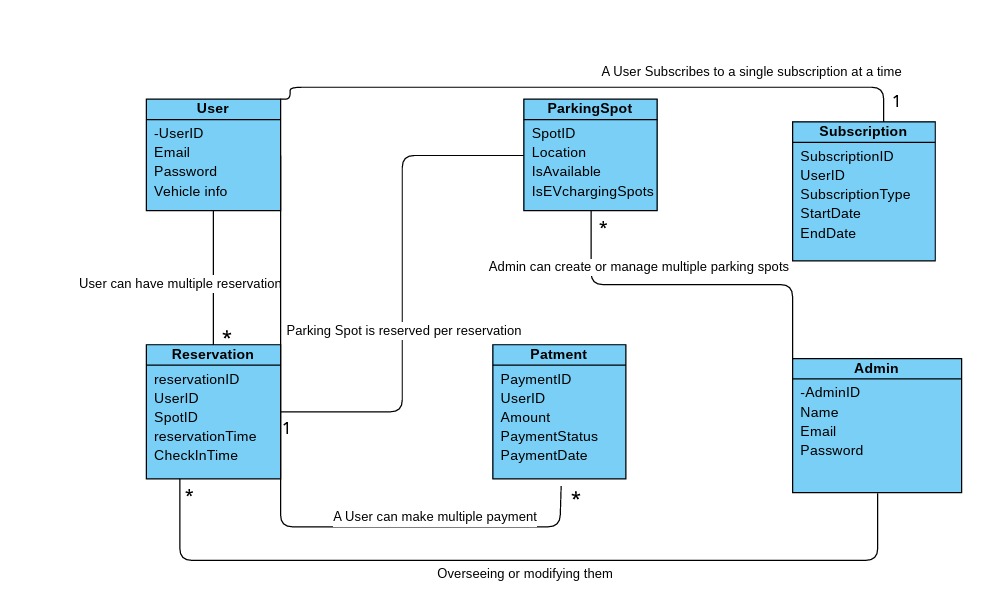
5. Class Diagram

Figure 4.5.1 Class Diagram

6. Entity Relationship Diagram

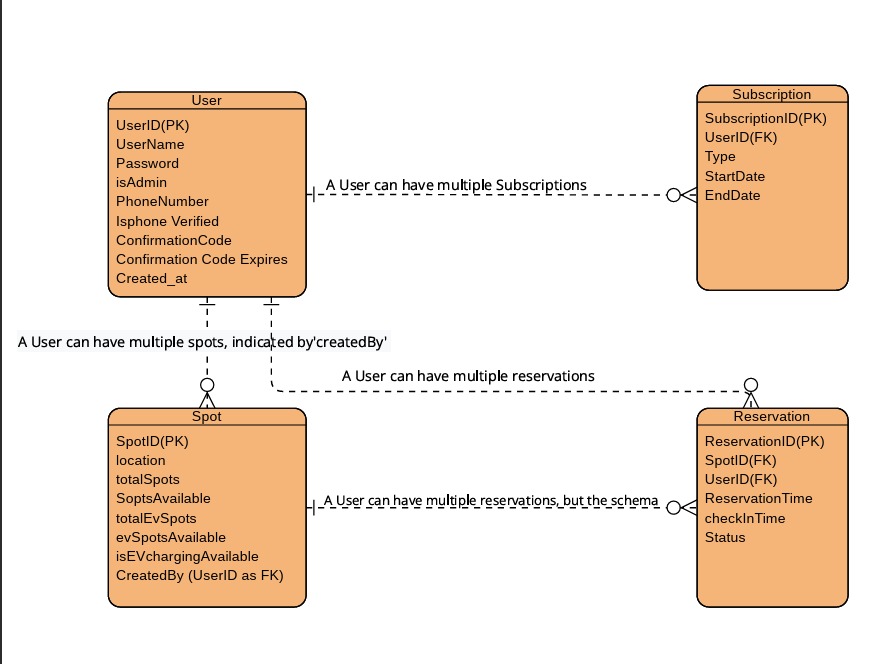


Figure 4.6.1 Entity Relationship Diagram

Chapter 5: Conclusion & Future Work

1.Summary of Achievements

The "ParkEase" project has successfully completed the initial design and documentation phases. Key achievements include:

* Comprehensive Documentation: Developed a thorough set of documents outlining system requirements, architecture, and design.
* System Architecture and Design: Created detailed system architecture and design diagrams to guide future development.
* User Interface Prototypes: Completed preliminary designs and user interface prototypes using Figma, ensuring a user-friendly experience.

2.Evaluation of Project Goals

The project's documentation phase was aimed at laying a robust groundwork for the development phase:

* Clarity and Precision: The documentation provides clear and precise guidelines for developers, which will facilitate the efficient development of the app.
* Preparation for Implementation: The groundwork has been prepared with a detailed project plan and technological framework, setting the stage for successful implementation.

3.Limitations and Challenges

The project faced several challenges during the documentation phase:

* Scope Definition: Defining the exact scope of the application required extensive research and adjustment to align with realistic goals.
* Technical Planning: Deciding on the optimal technologies and tools involved extensive deliberations to balance functionality, cost, and future scalability.

4.Recommendations for Future Work

For the next phases of the "ParkEase" project, the following steps are recommended:

* Begin Development: Transition from planning to the development phase, implementing the designs and plans laid out in the documentation.
* User Testing and Feedback: Early and continuous user testing to iterate on the design based on real-world feedback.
* Marketing and User Education: Develop a marketing plan to increase user adoption and educate potential users about the benefits of the app.

5.Concluding Remarks

"ParkEase" is well-positioned for the next stages of development and eventual deployment. The documentation created provides a strong foundation for building a solution that not only addresses the challenges of urban parking but also enhances the experience of city driving. Future development will be crucial in bringing the documented

**References**

Aftab, K., Kulkarni, P., Shergold, I., Jones, M., Dogramadzi, M., Carnelli, P., & Sooriyabandara, M. (2020). Reducing Parking Space Search Time and Environmental Impacts: A Technology Driven Smart Parking Case Study. IEEE Technology and Society Magazine, 39(3), 62-75. https://doi.org/10.1109/MTS.2020.3012329

Abbas, M. M., & Gabriel, K. R. (1998). Accurate differential global positioning system via fuzzy logic Kalman filter sensor fusion technique. IEEE Transactions on Industrial Electronics, 45(4), 572-579. https://doi.org/10.1109/41.704874

Cheng, Y., Jin, L., & Chu, L. (2010). Optimizing Freeway Traffic Sensor Locations by Clustering Global-Positioning-System-Derived Speed Patterns. IEEE Transactions on Intelligent Transportation Systems, 11(3), 714-724. https://doi.org/10.1109/TITS.2010.2047868